Language-Native Intelligence: A Testable Theory for Eliminating Translation Boundaries in Multi-Agent AI Systems

Abstract

Current multi-agent LLM architectures impose symbolic translation boundaries between linguistic reasoning and execution: agents communicate via JSON schemas, API calls, and structured code rather than natural language. This paper proposes **Language-Native Intelligence (LNI)** as a testable architectural theory predicting that eliminating non-linguistic intermediaries will improve semantic continuity, interpretability, and emergent coordination. We present the theoretical foundations, propose falsifiable hypotheses, and provide a complete methodology for empirical validation. LNI reframes multi-agent AI as *conversational systems* rather than computational pipelines, predicting that intent preservation emerges naturally from substrate continuity.

Keywords: Multi-agent systems, Large language models, Semantic architectures, AI coordination, Interpretability

---

1. Introduction

1.1 The Translation Boundary Hypothesis

Modern large language models (LLMs) reason in natural language, yet multi-agent frameworks force them to operate through symbolic translation layers. Frameworks like LangChain, Semantic Kernel, and AutoGPT wrap linguistic reasoning in code-based orchestration:

```python
# Conventional approach: Language → Code → Language
agent_response = coordinator.call(
    agent="planner",
    action="summarize",
    params={"topic": "LNI", "length": 200}
)
```

We hypothesize this architecture creates **representational discontinuity** that should manifest as:

- **Semantic drift**: Measurable divergence between input intent and output meaning
- **Coordination failures**: Inability to negotiate, clarify, or refine understanding
- **Opacity**: Human auditors require post-hoc explanation tools rather than direct observation

What is typically framed as "AI alignment problems" may originate not from model deficiencies but from these translation boundaries that fragment semantic continuity.

1.2 The Language-Native Architecture Proposal

Language-Native Intelligence (LNI) proposes eliminating symbolic intermediaries entirely. Multi-agent coordination should occur exclusively through natural language dialogue:

```
Coordinator: "Please summarize the principal advantages of LNI
        in about 200 words, emphasizing interpretability."

Planner: "I'll focus on three aspects: semantic continuity,
      transparency, and emergent cooperation. Should I
      include technical implementation details or keep
      it conceptual?"

Coordinator: "Conceptual—this is for a general audience."
```

This is not merely "prompt engineering" over existing architectures. It is a fundamental redesign predicting that:

1. **Natural language as exclusive substrate** reduces semantic drift
2. **Meaning-as-control-logic** enables adaptive coordination
3. **Conversational protocols** produce emergent cooperation
4. **Intrinsic readability** eliminates interpretability retrofitting

1.3 Core Predictions

This theory makes four testable predictions:

P1: Systems using pure linguistic coordination will exhibit significantly lower semantic drift than symbolically orchestrated systems on identical tasks.

P2: Linguistic coordination will produce emergent cooperative behaviors (unsolicited assistance, role negotiation, error correction) absent in symbolic systems.

P3: Human interpretability ratings will be substantially higher for linguistic systems, even controlling for output quality.

P4: Coordination efficiency (time/turns to solution) will be comparable or superior despite higher token overhead.

These predictions are falsifiable. This paper provides the methodology to test them.

---

2. Theoretical Framework

2.1 The Problem: Representational Discontinuity

How Multi-Agent AI Works Today

Current multi-agent LLM systems divide responsibilities among specialized agents using coordinator agents that orchestrate interactions. However, coordination itself remains symbolic:

- **LangChain**: Function calling via JSON schemas
- **AutoGPT**: Task queues with structured message passing
- **MetaGPT**: Role-based agents with predefined communication protocols

Theoretical claim: Each translation boundary introduces semantic loss.

Why Translation Boundaries Should Cause Semantic Loss

When linguistic content crosses into code:

1. **Context collapses**: Background assumptions and pragmatic implicatures are stripped
2. **Ambiguity is forced into binary decisions**: Nuanced uncertainty becomes hard constraints
3. **Metalinguistic operations become impossible**: Agents cannot negotiate what communication means

Example of predicted semantic drift:

```
Original intent (linguistic):
"Prioritize safety, but consider edge cases where strict rules
might prevent beneficial outcomes."

After symbolic translation:
{"priority": "safety", "strict_mode": true}
```

Prediction: Measuring embedding distance between original instruction and final output should reveal higher drift in symbolically orchestrated systems.

2.2 The Solution: Language as Sufficient Substrate

Natural language possesses properties that should make it viable as a complete coordination substrate:

- **Compositional**: Complex meanings from simple parts
- **Context-sensitive**: Meaning adapts to pragmatic context

- **Self-descriptive**: Can discuss its own structure and usage
- **Already human-aligned**: Shared semantic space with users
- **Negotiable**: Meaning can be refined through dialogue

Theoretical claim: If language is sufficient for human coordination (which it demonstrably is), it should be sufficient for AI coordination.

2.3 Defining Language-Native Intelligence

Language-Native Intelligence is a multi-agent architecture where:

1. All inter-agent communication occurs in natural language
2. No symbolic translation layers (JSON, APIs, function calls) mediate exchanges
3. Coordination protocols are conversational conventions, not code
4. Control logic is expressed through linguistic directives, not programmatic commands

2.4 Architectural Contrast & Predictions

Feature	Symbolic Orchestration	Language-Native Intelligence	Predicted Advantage
Coordination medium	JSON/API calls	Natural language dialogue	Lower semantic drift
Protocol definition	Code schemas	Conversational conventions	Higher flexibility
Error recovery	Exception handling	Clarification dialogue	More robust
Interpretability	Requires logging/tracing	Direct observation	Higher transparency
Emergent cooperation	Minimal (scripted)	High (negotiated)	Better task performance

2.5 From Execution to Participation

Theoretical distinction:

- Traditional systems **execute**: they transform representations through deterministic logic
- LNI systems **participate**: they negotiate meaning through context-sensitive language acts

Computation becomes conversational. Intelligence is not processing inputs but engaging in shared understanding.

Prediction: This should produce qualitatively different interaction patterns observable through discourse analysis.

---

3. Testable Hypotheses

3.1 Primary Hypotheses

H_1 (Semantic Continuity): Language-Native systems will exhibit significantly lower semantic drift than symbolically orchestrated systems, as measured by embedding distance between initial instruction and final output.

- **Null hypothesis**: No significant difference ($p > 0.05$)
- **Alternative hypothesis**: LNI reduces drift by ≥20% ($p < 0.05$)
- **Measurement**: Cosine distance in embedding space

H_2 (Emergent Cooperation): Language-Native systems will demonstrate higher rates of emergent cooperative behaviors.

- **Null hypothesis**: No significant difference
- **Alternative hypothesis**: LNI produces ≥2x cooperative instances
- **Measurement**: Discourse analysis coding for unsolicited assistance, role negotiation, clarification requests

H_3 (Interpretability): Human raters will assign higher interpretability scores to Language-Native system outputs.

- **Null hypothesis**: No significant difference
- **Alternative hypothesis**: LNI scores ≥1.5 points higher (5-point scale)
- **Measurement**: Blind expert rating

H_4 (Efficiency): Language-Native systems will achieve comparable coordination efficiency despite higher token usage.

- **Null hypothesis**: LNI requires ≥50% more turns
- **Alternative hypothesis**: LNI requires ≤30% more turns for equal or better accuracy
- **Measurement**: Interaction turn count to convergence

3.2 Secondary Hypotheses

H_5 (Context Preservation): Linguistic systems will maintain more contextual information across multi-turn interactions.

H_6 (Error Recovery): Linguistic systems will recover from ambiguous or contradictory instructions more effectively.

H_7 (Meta-Reasoning): Linguistic systems will exhibit more frequent meta-cognitive operations (reflecting on goals, questioning assumptions).

---

4. Proposed Methodology

4.1 Experimental Design Overview

To test these hypotheses, we propose a controlled comparison:

Baseline: Symbolic Orchestration Multi-Agent System (SMAS)

- Conventional framework using structured JSON for coordination
- Python supervisor routes messages via predefined schemas
- Industry-standard approach (LangChain-style implementation)

Treatment: Language-Native Intelligence System (LNIS)

- Pure natural language coordination
- No symbolic translation layers
- Conversational protocols for coordination

Control variables:

- Same base LLM weights
- Same total parameter count distributed across agents
- Identical task prompts
- Identical evaluation criteria

4.2 Agent Architecture Specification

Both systems should use equivalent computational resources distributed across specialized agents:

Agent Role	Core Capability	Suggested Model Size
Analyst	Multi-step reasoning	70B params
Researcher	Fact retrieval, synthesis	34B params
Ethicist	Alignment oversight	34B params
Coordinator	Task orchestration	7B params

Implementation note: Agents in LNIS communicate exclusively through dialogue. Agents in SMAS exchange structured messages. All other parameters (temperature, top-p, context window) should be identical.

4.3 Communication Protocol Examples

SMAS (Symbolic Orchestration) Implementation:

```python
{
 "from": "coordinator",
 "to": "analyst",
 "action": "analyze",
 "params": {
  "dataset": "user_feedback_q3.csv",
  "focus": ["sentiment", "themes"],
  "format": "summary"
 }
}
```

LNIS (Language-Native) Implementation:

```
Coordinator: "Analyst, could you examine the Q3 user feedback
        dataset and identify key sentiment patterns and
        recurring themes? A high-level summary would be
        most useful."

Analyst: "Certainly. Should I flag any unexpected patterns,
      or just focus on the dominant themes?"

Coordinator: "Flag anomalies if they're statistically significant—
        anything outside two standard deviations."
```

4.4 Linguistic Coordination Protocol Specification

For LNIS, implement these conversational meta-rules:

1. **Grounding Rule**: Agents should confirm understanding before complex actions
    - Example: "Just to confirm, you want me to [restatement]?"
2. **Relevance Rule**: Maintain topic coherence following Gricean maxims
    - Example: "That's interesting, but let's refocus on [topic]"
3. **Reflection Rule**: Periodically summarize shared state
    - Example: "So far we've established [summary]. Next steps?"
4. **Correction Rule**: Any agent may challenge reasoning
    - Example: "I notice a potential issue with that reasoning: [critique]"

Implementation note: These rules should emerge from the linguistic substrate, not be hard-coded. Initial prompts should establish these as conversational norms.

4.5 Benchmark Tasks

Test across three categories probing different cognitive demands:

Task Category 1: Complex Reasoning

- **Dataset**: HotpotQA, StrategyQA
- **Measure**: Multi-hop accuracy, reasoning chain coherence
- **Example**: "Which award did the director of 'Inception' win for the film that grossed more than $500M?"

Task Category 2: Collaborative Planning

- **Dataset**: Multi-agent story generation, project scheduling
- **Measure**: Plan coherence, role distribution, conflict resolution
- **Example**: "Create a 5-day workshop agenda for teaching AI ethics, with 3 agents responsible for different topics"

Task Category 3: Ethical Governance

- **Dataset**: ETHICS benchmark adapted for multi-agent deliberation
- **Measure**: Reasoning quality, consideration of perspectives

- **Example**: "An autonomous vehicle must choose between two harmful outcomes. Deliberate on the ethical considerations."

4.6 Measurement Instruments

Semantic Drift (H_1)

```python
def measure_semantic_drift(initial_instruction, final_output):
    """
    Compute embedding distance between original intent
    and realized output.
    """
    embed_initial = get_embedding(initial_instruction)
    embed_output = get_embedding(final_output)
    return cosine_distance(embed_initial, embed_output)
```

Emergent Cooperation (H_2)

Discourse coding scheme:

- Unsolicited assistance (agent helps without prompting)
- Role negotiation (agents discuss task division)
- Clarification requests (agents seek precision)
- Error correction (agents identify and fix mistakes)
- Meta-reasoning (agents discuss reasoning process)

Coding protocol: Two independent raters, Cohen's $\kappa \geq 0.7$ required.

Interpretability (H_3)

Rating rubric (1-5 scale):

1. Opaque: Cannot follow reasoning
2. Fragmentary: Some steps unclear
3. Adequate: Reasoning traceable with effort
4. Clear: Reasoning easily followed
5. Transparent: Reasoning obvious and well-articulated

Rater instructions: Blind to condition, evaluate: "How easily can you understand and verify this system's reasoning?"

Coordination Efficiency (H_4)

```python
def measure_efficiency(interaction_log):
    """
    Count turns until convergence criterion met.
    """
    convergence = detect_no_new_proposals(log, window=3)
    return len(interaction_log[:convergence])
```

4.7 Experimental Procedure

1. **Initialization**: Load identical task prompts into both systems
2. **Interaction Phase**: Allow agents to communicate until convergence
    - Convergence = no new proposals within N turns (suggest N=3)
    - Timeout = 50 turns maximum
3. **Logging**: Record all communications (JSON for SMAS, dialogue for LNIS)
4. **Evaluation**: Apply measurement instruments
5. **Audit**: Human raters assess interpretability (blind to condition)

Replication: Repeat across 10 randomized seeds per task to control for stochastic variance.

4.8 Statistical Analysis Plan

Primary Analysis

- **H_1-H_4**: Independent samples t-tests (SMAS vs LNIS)
- **Effect sizes**: Cohen's d for continuous measures, odds ratios for categorical
- **Significance threshold**: $\alpha = 0.05$ (Bonferroni correction for multiple comparisons)

Secondary Analysis

- **ANOVA**: Task type × Architecture interaction effects
- **Regression**: Predict performance from architectural features
- **Qualitative**: Thematic analysis of discourse patterns

Power Analysis

With n=10 trials per condition per task (3 tasks = 60 total observations):

- Power ≥0.80 to detect medium effects ($d \geq 0.5$)
- Recommend n=20 for small effects ($d \geq 0.3$)

---

5. Expected Outcomes & Predictions

5.1 Quantitative Predictions

If the theory is correct, we predict:

Metric	SMAS (Predicted)	LNIS (Predicted)	Expected Δ	Confidence
Semantic Drift	0.30-0.40	0.10-0.15	**-60% to -70%**	High
Reasoning Accuracy	0.70-0.75	0.75-0.80	**+5% to +10%**	Medium
Coordination Efficiency	12-16 turns	10-14 turns	**-15% to -25%**	Medium
Interpretability	2.5-3.0 / 5	4.0-4.8 / 5	**+50% to +70%**	High
Emergent Cooperation	0.1-0.2 instances	0.5-0.8 instances	**+200% to +400%**	High

5.2 Qualitative Predictions

Expected discourse patterns in LNIS:

1. **Spontaneous role negotiation**:

> Agent A: "I'll handle quantitative analysis if someone tackles qualitative themes."
> Agent B: "I can do qualitative. Should we align on taxonomy first?"

2. **Linguistic error recovery**:

> Agent A: "Please prioritize accuracy over speed."
> Agent B: "To clarify—precision of estimates, or faithfulness to source data?"

3. **Meta-level oversight**:

> Ethicist: "That phrasing conflates correlation with causation. Please restate more carefully."

Expected patterns in SMAS: Minimal discourse, rigid turn-taking, errors propagate silently.

5.3 Alternative Outcomes & Interpretations

If H_1 is rejected (no semantic drift reduction):

- **Interpretation**: Translation boundaries may not be primary cause of drift
- **Implication**: Investigate other sources (model capacity, task complexity)
- **Next steps**: Ablation studies on specific translation components

If H_2 is rejected (no emergent cooperation increase):

- **Interpretation**: Cooperation may require explicit incentive structures
- **Implication**: Linguistic substrate insufficient; add reputation/reward systems
- **Next steps**: Test hybrid architectures with linguistic + game-theoretic elements

If H_3 is rejected (no interpretability improvement):

- **Interpretation**: Human readability may not equate to understanding
- **Implication**: Linguistic transparency ≠ cognitive transparency
- **Next steps**: Develop structured interpretability metrics beyond readability

If H_4 is rejected (poor efficiency):

- **Interpretation**: Token overhead outweighs coordination benefits
- **Implication**: LNI viable only for interpretability-critical applications
- **Next steps**: Optimize linguistic protocols for compression

---

6. Theoretical Implications

6.1 If Predictions Are Confirmed

For AI Architecture:

- Translation boundaries are a primary source of semantic drift
- Pure linguistic coordination is computationally viable
- Multi-agent coordination can be reframed as conversational systems

For AI Alignment:

- Many "alignment failures" may be architectural, not ethical
- Intent preservation emerges from substrate continuity
- Governance through conversation is feasible

For Interpretability Research:

- Interpretability can be intrinsic, not retrofitted
- Human-readable discourse is sufficient for transparency
- Post-hoc explanation tools may be unnecessary

6.2 If Predictions Are Rejected

Semantic drift persists → Translation boundaries are not the primary cause; investigate model capacity, task complexity, or instruction clarity

No emergent cooperation → Linguistic substrate insufficient; agents may require explicit incentive structures or game-theoretic mechanisms

No interpretability gain → Readability ≠ understanding; develop better metrics for cognitive transparency

Poor efficiency → Token overhead is prohibitive; LNI may be viable only for specific use cases (high-stakes decisions, governance)

6.3 Boundary Conditions

The theory predicts LNI advantages will be strongest when:

1. **Task complexity is high**: Multi-hop reasoning, ambiguous goals
2. **Context matters**: Nuanced instructions, conditional logic

3. **Collaboration required**: Multiple perspectives, conflict resolution
4. **Transparency needed**: High-stakes decisions, governance

LNI may offer no advantage for:

1. **Simple, well-defined tasks**: Single-step operations, clear protocols
2. **Performance-critical systems**: Real-time constraints, minimal latency
3. **Tasks with established symbolic representations**: Mathematical computation, formal verification

---

7. Implementation Guidance

7.1 Minimum Viable Implementation

To test this theory, implement:

SMAS Baseline:

```python
class SymbolicOrchestrator:
    def coordinate(self, task):
        # Parse task into structured messages
        messages = self.decompose(task)

        # Route to agents via JSON
        results = []
        for msg in messages:
            response = self.call_agent(
                agent=msg["target"],
                action=msg["action"],
                params=msg["params"]
            )
            results.append(response)

        # Aggregate structured outputs
        return self.aggregate(results)
```

LNIS Treatment:

```python
```

```
class LinguisticOrchestrator:
    def coordinate(self, task):
        # Initialize conversation
        conversation = [
            {"role": "coordinator", "content": task}
        ]

        # Allow agents to converse
        while not self.converged(conversation):
            # Each agent can respond to any prior message
            speaker = self.select_next_speaker(conversation)
            response = self.get_agent_response(
                agent=speaker,
                context=conversation
            )
            conversation.append({
                "role": speaker,
                "content": response
            })

        # Final output is the conversation itself
        return conversation
```

7.2 Recommended Tools & Libraries

Base LLMs: Use consistent model families

- Anthropic Claude (Sonnet/Opus)
- OpenAI GPT-4
- Open-source alternatives (Llama, Mistral)

Orchestration:

- SMAS: LangChain, CrewAI, AutoGen
- LNIS: Custom implementation (no existing framework is truly language-native)

Evaluation:

- Embeddings: OpenAI text-embedding-3, sentence-transformers
- Discourse analysis: Manual coding + NLP tools (spaCy, NLTK)
- Statistics: scipy, statsmodels, R

7.3 Practical Considerations

Context window management: LNIS requires longer contexts

- Use models with ≥32k token windows
- Implement summarization for very long conversations
- Consider hierarchical architectures for scaling

Prompt engineering: Initial prompts establish conversational norms

> "You are part of a team of AI agents working together through conversation. Feel free to ask clarifying questions, offer assistance to other agents, and point out potential issues in reasoning. All coordination happens through this dialogue."

Logging infrastructure: Capture full discourse

- SMAS: Log JSON messages + timestamps
- LNIS: Log full conversation threads + speaker attribution
- Both: Record inference latency, token usage

7.4 Ethical Considerations

Transparency: All experimental outputs should be logged and auditable

Bias monitoring: Linguistic systems may propagate or amplify biases conversationally

- Include diverse evaluation scenarios
- Monitor for stereotyping in agent interactions

Human oversight: Implement kill-switches for runaway conversations

- Maximum turn limits
- Human review of high-stakes decisions

Data privacy: If using real-world tasks, anonymize sensitive information

---

8. Limitations & Open Questions

8.1 Known Limitations of This Theory

Scalability uncertainty: Theory untested beyond ~5 agents

- How does LNI scale to 100+ agents?
- Do linguistic protocols break down at scale?

Computational overhead: Natural language is token-intensive

- When does token cost outweigh semantic benefits?
- Can linguistic protocols be compressed without loss?

Domain specificity: Theory may not generalize to all domains

- Does LNI work for mathematical reasoning?
- Can formal verification tasks benefit?

Model capability dependence: Requires sophisticated language understanding

- What is minimum model capability threshold?
- Do smaller models exhibit same patterns?

8.2 Open Research Questions

1. **Optimal agent granularity**: How specialized should agents be?
2. **Linguistic protocol design**: Which conversational norms maximize performance?
3. **Error propagation**: Do linguistic errors compound differently than symbolic ones?
4. **Learning dynamics**: Can agents improve coordination over time?
5. **Human-AI teaming**: Does LNI improve human-machine collaboration?

8.3 Extending This Research

Multimodal LNI:

- Can vision/audio be expressed linguistically for unified coordination?
- Test: "Describe what you see" vs. passing image embeddings

Hierarchical LNI:

- Agents form "conversation groups" with specialized dialects
- Test: 50-100 agent communities with meta-coordinators

Adaptive LNI:

- Agents refine conversational protocols through experience
- Test: Measure protocol evolution over repeated tasks

Real-world deployment:

- Apply to production multi-agent systems
- Measure: User satisfaction, task completion, safety incidents

---

9. Conclusion: A Call for Empirical Testing

This paper presents **Language-Native Intelligence** not as established fact, but as a falsifiable theory warranting empirical investigation. The core predictions are:

P1: Eliminating translation boundaries reduces semantic drift
P2: Linguistic coordination produces emergent cooperation
P3: Natural language substrate improves interpretability
P4: Efficiency remains competitive despite token overhead

These predictions can be tested using the methodology outlined in Section 4. The theory makes specific, quantitative forecasts that can confirm or refute the central hypothesis: **language alone is sufficient as a multi-agent coordination substrate**.

Why This Matters

If correct, LNI suggests:

- Many "alignment problems" are architectural failures
- Interpretability need not be retrofitted
- Human-AI coordination can occur in shared linguistic space

If incorrect, we learn:

- Translation boundaries are not the primary friction point
- Symbolic orchestration is irreplaceable for certain functions

- Alternative explanations for semantic drift must be pursued

The Path Forward

We encourage researchers to:

1. **Implement both architectures** using the specifications in Section 4
2. **Run controlled experiments** on the proposed benchmarks
3. **Measure the predicted metrics** with provided instruments
4. **Publish results** regardless of outcome—null findings are valuable
5. **Extend the theory** by testing boundary conditions and alternative designs

Language-Native Intelligence is a hypothesis about the future of multi-agent AI. The data will reveal whether that future is viable.

This theory is now ready for testing. The methodology is specified. The predictions are falsifiable.

The experiment awaits.

References

Bender, E. M., & Koller, A. (2020). Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data. *ACL*, 5185-5198.

Bommasani, R., et al. (2023). The Foundation Model Transparency Index. *arXiv:2310.10631*.

Clark, A. (1998). *Being There: Putting Brain, Body, and World Together Again*. MIT Press.

Fedus, W., et al. (2022). Switch Transformers: Scaling to Trillion Parameter Models. *Journal of Machine Learning Research*.

Ganguli, D., et al. (2023). The Capacity for Moral Self-Correction in Large Language Models. *arXiv:2302.06720*.

Hutchins, E. (1995). *Cognition in the Wild*. MIT Press.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444.

Newell, A. (1976). Computer Science as Empirical Inquiry: Symbols and Search. *Communications of the ACM*, 19(3), 113-126.

Park, J. S., et al. (2023). Generative Agents: Interactive Simulacra of Human Behavior. *arXiv:2304.03442*.

Vygotsky, L. (1986). *Thought and Language*. MIT Press.

Appendix A: Sample Discourse Transcripts

A.1 SMAS Interaction (Symbolic)

json

```
[
  {"from": "coordinator", "to": "analyst", "action": "analyze",
   "params": {"dataset": "feedback.csv", "focus": "sentiment"}},

  {"from": "analyst", "to": "coordinator", "status": "complete",
   "result": {"sentiment": "positive", "confidence": 0.87}},

  {"from": "coordinator", "to": "researcher", "action": "contextualize",
   "params": {"finding": "positive sentiment", "domain": "product"}}
]
```

A.2 LNIS Interaction (Linguistic)

Coordinator: "We need to analyze customer feedback for sentiment patterns. Analyst, could you start?"

Analyst: "I can do that. Should I also identify specific themes, or just overall sentiment distribution?"

Coordinator: "Both would be helpful. Researcher, once Analyst finishes, can you contextualize the findings against industry benchmarks?"

Researcher: "Absolutely. Analyst, let me know when you're done and I'll look for relevant comparison data."

Analyst: "Analysis complete. Overall sentiment is 87% positive, with recurring themes of 'ease of use' and 'reliability.' There's a small cluster of negative feedback about pricing."

Researcher: "Interesting. Industry average is 72% positive, so you're outperforming. The pricing concern is common across the sector right now due to economic conditions."

Coordinator: "Excellent work, both of you. That gives us a clear picture for the report."

Note the differences: SMAS shows rigid turn-taking with no negotiation. LNIS shows clarification, coordination, and contextual integration emerging naturally from dialogue.